

Программное обеспечение
«Servicepipe Application Delivery Controller»
Инструкция по установке программного обеспечения

Москва — 2026 г.

Оглавление

1 Общие сведения	4
1.1 Назначение документа	4
1.2 Сведения о программном обеспечении	4
1.3 Область применения инструкции	4
1.4 Термины и сокращения	4
2 Модель установки и состав поставки	5
2.1 Общая модель online-установки	5
2.2 Состав установочного комплекта	5
2.3 Основные компоненты установки	6
2.4 Роли узлов и пример топологии	6
3 Предварительные требования	7
3.1 Требования к controller-хосту	7
3.2 Требования к целевым хостам	7
3.3 Сетевой доступ к репозиториям	7
3.4 Данные, необходимые перед началом установки	8
4 Подготовка установочного комплекта	8
4.1 Получение и распаковка online-архива	8
4.2 Проверка рабочей директории установщика	8
5 Первичная настройка установщика	9
5.1 Назначение bootstrap	9
5.2 Запуск bootstrap	9
5.3 Vault password и сгенерированные секреты	9
5.4 Настройка топологии и состава компонентов	9
5.5 Пример выполнения bootstrap	9
5.6 Проверка inventory после bootstrap	10
6 Установка программного обеспечения	10
6.1 Установка всех выбранных компонентов	10
6.2 Установка отдельных компонентов	11
6.3 Повторный запуск установки и обработка ошибок	11
7 Проверка результата установки	11
7.1 Проверка системных сервисов	11
7.2 Проверка PostgreSQL HA и Patroni	12
7.3 Проверка VictoriaMetrics	12
7.4 Проверка контейнерного контура личного кабинета	12
7.5 Критерии успешного завершения установки	12
8 Обновление программного обеспечения	13
8.1 Общий порядок обновления	13
8.2 Подготовка нового online-архива	13
8.3 Обновление всех компонентов	13
8.4 Обновление одного компонента	14
8.5 Обновление только container images личного кабинета	14

9 Удаление программного обеспечения	14
9.1 Общие правила удаления	14
9.2 Удаление компонентов	14
9.3 Удаление с данными и пакетами	14
9.4 Рекомендуемый порядок удаления	15
10 Типовые проблемы и способы устранения	15
10.1 Для sudo требуется пароль	15
10.2 Хост завершился с ошибкой, следующие этапы пропущены	15
10.3 DNF не находит пакет	15
10.4 Не скачивается архив из raw repository	15
10.5 Неверный vault password	16
11 Завершение работ и хранение установочных материалов	16

1 Общие сведения

1.1 Назначение документа

Настоящий документ содержит инструкцию по установке программного обеспечения «Servicepipe Application Delivery Controller» (далее - ПО «Servicepipe ADC», ADC, программное обеспечение) в инфраструктуре заказчика или private cloud.

Документ описывает подготовку установочного комплекта, выполнение первичной настройки установщика, запуск установки, проверку результата, порядок обновления и удаления компонентов программного обеспечения.

Инструкция подготовлена для администраторов, инженеров эксплуатации, DevOps/SRE-специалистов и иных уполномоченных сотрудников, выполняющих развертывание и сопровождение Servicepipe ADC.

1.2 Сведения о программном обеспечении

ПО «Servicepipe ADC» предназначено для публикации прикладных сервисов и API, маршрутизации и балансировки L4/L7-трафика, включая TCP/UDP и HTTP/HTTPS, терминирования и проксирования TLS-соединений, а также централизованного управления конфигурацией.

Программное обеспечение размещается перед backend-сервисами заказчика, принимает входящий трафик, сопоставляет его с правилами обработки, применяет политики TLS, маршрутизации, балансировки, доступности, логирования и обработки HTTP-трафика, после чего передает запрос или соединение в соответствующий upstream.

Управление функциями ADC выполняется через административный контур и API управления. В состав поставки входят компоненты runtime-слоя, компоненты личного кабинета и API управления, инфраструктурные сервисы хранения конфигурации, метрик и журналов, а также установочные сценарии, необходимые для развертывания продукта.

1.3 Область применения инструкции

Настоящая инструкция описывает online-установку Servicepipe ADC из готового online installer archive. Такой архив содержит установщик, inventory-шаблоны, playbook-и, переменные и служебные файлы, но не содержит полный набор RPM-пакетов, архивов сервисов и дополнительных бинарных файлов компонентов.

В процессе установки целевые хосты получают RPM-пакеты из настроенного DNF repository, а недостающие архивы, файлы поставки и container images скачиваются из gaw repository или иного хранилища артефактов, указанного в параметрах установщика.

1.4 Термины и сокращения

Термин	Описание
Controller-хост	Хост, с которого оператор запускает Ansible-установщик и выполняет команды установки, обновления и удаления. Не обязательно входит в состав целевого кластера ADC.
Целевой хост	Сервер инфраструктуры заказчика, на который устанавливаются компоненты Servicepipe ADC.

Mgmt nodes	Группа управляющих или инфраструктурных узлов, на которых размещаются компоненты управления, хранения данных, метрик, журналов и служебные сервисы.
Filter nodes / runtime nodes	Узлы обработки трафика и смежных функций, на которых могут размещаться runtime-компоненты, агенты метрик и иные сервисы, связанные с обработкой или наблюдаемостью трафика.
Online installer archive	Архив установщика, предназначенный для online-развертывания. В текущей схеме поставки используется имя вида <code>cresp-onprem-ansible-online-<version>.tar.gz</code> .
DNF repository	Репозиторий RPM-пакетов, из которого целевые хосты получают пакеты во время установки и обновления.
Raw repository	Хранилище архивов, файлов поставки и container images, которые не входят в online installer archive и скачиваются во время установки.
Vault password	Пароль Ansible Vault, используемый для шифрования и чтения сгенерированных секретов установщика.
Playbook	Ansible-сценарий, выполняющий установку, обновление или удаление конкретного компонента либо всего комплекта.

2 Модель установки и состав поставки

2.1 Общая модель online-установки

Online-установка Servicepipe ADC выполняется с controller-хоста с использованием Ansible. Оператор распаковывает online installer archive, запускает bootstrap для формирования рабочей конфигурации установщика, после чего выполняет один или несколько Ansible playbook-ов.

Процесс установки включает следующие основные этапы:

- подготовка controller-хоста и проверка доступа к целевым хостам;
- проверка доступности DNF repository и raw repository;
- распаковка online installer archive;
- первичная настройка установщика с помощью bootstrap.sh;
- формирование inventory, group vars, vault vars и команд запуска;
- запуск установки всех выбранных компонентов или отдельных компонентов;
- проверка состояния системных сервисов, контейнеров и служебных HTTP-endpoint-ов;
- сохранение установочных материалов и сведений о выполненной установке.

2.2 Состав установочного комплекта

Online installer archive содержит файлы, необходимые для управления процессом установки, но не является полным offline-репозиторием артефактов. Конкретный состав архива зависит от версии поставки, но в общем случае включает:

- скрипт bootstrap.sh для первичной настройки установщика;
- Ansible playbook-и для установки, обновления и удаления компонентов;

- inventory-шаблоны и переменные групп хостов;
- шаблоны конфигурационных файлов;
- служебный файл `.installer.env` с параметрами репозиторий и именами артефактов;
- manifest версии и сведения о составе комплекта поставки;
- документацию и вспомогательные файлы установщика.

В текущей схеме поставки в примерах используется каталог установщика `cresp-onprem-ansible`. Если в конкретной поставке каталог или архив имеют другое имя, необходимо использовать фактическое имя из `release manifest` и комплекта поставки.

2.3 Основные компоненты установки

Состав устанавливаемых компонентов выбирается во время `bootstrap` и определяется топологией стенда. Ниже приведен типовой состав компонентов, используемых Servicepipe ADC.

Компонент	Назначение
ClickHouse cluster	Хранение и обработка журналов, аналитических данных и иных событий, используемых для диагностики и наблюдаемости.
Kafka stack	Брокер событий и журналов, используемый для передачи потоков данных между компонентами системы.
VictoriaMetrics stack	Хранилище метрик, используемое для сбора и анализа показателей работы компонентов ADC.
vmagent	Агент доставки метрик в VictoriaMetrics, в том числе с <code>filter/runtime nodes</code> .
Vector	Агент обработки и доставки журналов. Может размещаться на <code>mgmt nodes</code> и использоваться для передачи логов в целевые хранилища.
PostgreSQL HA cluster	Отказоустойчивый кластер базы данных для компонентов управления, личного кабинета и API.
LK docker stack	Контейнерный контур личного кабинета и API управления Servicepipe ADC, включая <code>frontend/backend</code> -компоненты управления.
Local-repo client configuration	Настройка доступа целевых хостов к используемым репозиториям пакетов и артефактов.

Примечание. Фактический перечень компонентов может быть расширен или сокращен в зависимости от версии Servicepipe ADC, выбранного режима эксплуатации и состава поставки. Приоритетным источником состава версии является `release manifest`, входящий в комплект поставки.

2.4 Роли узлов и пример топологии

Перед установкой необходимо определить роли узлов и их сетевые адреса. В минимальной инструкции используются следующие группы узлов: `mgmt nodes` и `filter nodes`. `Mgmt nodes` применяются для инфраструктурных компонентов и контура управления. `Filter nodes` используются для компонентов, связанных с обработкой трафика и сбором метрик на узлах обработки.

В примерах используются условные адреса:

```
mgmt1 10.250.10.11
mgmt2 10.250.10.12
mgmt3 10.250.10.13
filter1 10.250.20.11
```

```
filter2 10.250.20.12
filter3 10.250.20.13
```

Примечание. Перед использованием инструкции замените все условные адреса, имена пользователей, версии архивов, URL репозитория и имена компонентов на фактические значения, утвержденные для конкретного стенда.

3 Предварительные требования

3.1 Требования к controller-хосту

Перед запуском установки на controller-хосте должны быть выполнены следующие условия:

- получен готовый архив `cresp-onprem-ansible-online-<version>.tar.gz` или архив с фактическим именем, указанным в комплекте поставки;
- установлены `ansible`, `python3` и `ssh`;
- имеется SSH-доступ до всех целевых хостов;
- SSH-пользователь, используемый для установки, может выполнять `sudo` на целевых хостах;
- controller-хост имеет доступ к DNF repository и raw repository, если скачивание артефактов выполняется через controller-хост;
- известен vault password, который будет использоваться для сгенерированных секретов;
- доступно достаточное дисковое пространство для распаковки архива, хранения inventory, vault-файлов, временных файлов и журналов выполнения установки.

3.2 Требования к целевым хостам

На целевых хостах должны быть выполнены следующие условия:

- операционная система и системные пакеты соответствуют требованиям конкретной версии Servicepipe ADC;
- настроен сетевой доступ между узлами в соответствии с выбранной топологией;
- SSH-сервис доступен с controller-хоста;
- пользователь установки имеет права `sudo` либо оператор готов вводить пароль `sudo` при запуске `playbook`-ов;
- целевые хосты имеют доступ к DNF repository и raw repository, если артефакты скачиваются непосредственно с целевых хостов;
- системное время синхронизировано, так как корректная синхронизация времени важна для TLS, журналов, метрик и корреляции событий;
- на узлах подготовлены сетевые интерфейсы, маршруты, DNS-резолвинг и firewall-правила, необходимые для работы компонентов.

3.3 Сетевой доступ к репозиториям

В режиме `online` целевые хосты получают RPM-пакеты и дополнительные артефакты во время выполнения установки. До запуска `bootstrap` и `playbook`-ов необходимо проверить доступность используемых репозитория.

Проверка доступности DNF repository может выполняться следующими командами на целевых хостах:

```
dnf clean all
dnf makecache
dnf repoquery <package-name>
```

Проверка доступности raw repository может выполняться следующей командой:

```
curl -I <raw-repository-url>/<file-name>
```

3.4 Данные, необходимые перед началом установки

До начала установки рекомендуется подготовить и согласовать следующие данные:

- версию устанавливаемого комплекта Servicepipe ADC;
- имя и путь к online installer archive;
- состав компонентов, которые должны быть установлены;
- список mgmt nodes и filter/runtime nodes;
- SSH-пользователя для установки;
- порядок использования sudo: без пароля или с вводом пароля;
- URL DNF repository и raw repository;
- vault password для шифрования сгенерированных секретов;
- сетевые параметры стенда, включая адреса, DNS-имена, VIP-адреса, маршруты и правила доступа;
- контакт ответственного администратора и порядок фиксации результата установки.

4 Подготовка установочного комплекта

4.1 Получение и распаковка online-архива

Скопируйте online installer archive на controller-хост и распакуйте его в рабочий каталог, из которого будет выполняться установка.

```
tar -xzf cresp-onprem-ansible-online-<version>.tar.gz
cd cresp-onprem-ansible
```

Если архив или рабочий каталог имеют другое имя, используйте фактические значения из комплекта поставки. Все последующие команды должны выполняться из рабочей директории установщика, если не указано иное.

4.2 Проверка рабочей директории установщика

После распаковки проверьте наличие основных файлов и каталогов установщика. Типовая структура рабочей директории включает bootstrap.sh, каталог playbooks, каталог inventory, шаблоны переменных и служебный файл .installer.env.

```
ls -la
ls -la playbooks
ls -la inventory
```

Перед внесением изменений в конфигурацию рекомендуется сохранить исходное состояние распакованного каталога либо зафиксировать контрольную сумму полученного архива.

5 Первичная настройка установщика

5.1 Назначение bootstrap

Bootstrap подготавливает рабочие файлы установщика: inventory, group vars, vault vars, playbook-и для выбранных компонентов, секреты и команды запуска. Bootstrap необходимо запускать перед первой установкой.

Повторный запуск bootstrap требуется только при изменении топологии, состава компонентов, базовых настроек установки или при обновлении служебных параметров установщика.

5.2 Запуск bootstrap

Для запуска первичной настройки выполните команду из рабочей директории установщика:

```
./bootstrap.sh
```

В начале bootstrap оператор вводит vault password. Этот пароль сохраняется локально в файле .ansible/vault-password и используется для шифрования сгенерированных vault-файлов.

5.3 Vault password и сгенерированные секреты

Vault password является чувствительным значением. Доступ к файлу .ansible/vault-password должен быть ограничен только уполномоченными администраторами. Не рекомендуется передавать этот файл по незащищенным каналам связи или хранить его в общедоступных каталогах.

Если vault password утрачен, прочитать ранее сгенерированные vault-файлы стандартными средствами Ansible Vault будет невозможно. Поэтому пароль и порядок его хранения должны быть определены до начала установки.

5.4 Настройка топологии и состава компонентов

Во время bootstrap установщик задает вопросы о режиме установки, необходимости настройки online DNF repository и составе компонентов. Для типовой установки могут быть выбраны ClickHouse cluster, Kafka stack, VictoriaMetrics stack, vmagent, Vector, PostgreSQL HA cluster и LK docker stack.

Если для нескольких компонентов используются общие mgmt nodes, оператор может определить их один раз и затем использовать как источник узлов для компонентных блоков. Аналогично можно определить общие filter nodes для компонентов, которые устанавливаются на узлы обработки трафика или должны собирать с них метрики.

5.5 Пример выполнения bootstrap

Ниже приведен пример начала bootstrap и одного компонентного блока. Остальные выбранные компоненты настраиваются аналогично.

```
$ ./bootstrap.sh
Vault password for generated vault files:
=====
on-prem Ansible bootstrap
=====
```

```

Installation mode: online
Do you need to configure online DNF repository? [yes]:
Do you need ClickHouse cluster? [yes]:
Do you need Kafka stack? [yes]:
Do you need VictoriaMetrics stack? [yes]:
Do you need vmagent on filter nodes? [yes]:
Do you need Vector on mgmt nodes? [yes]:
Do you need PostgreSQL HA cluster? [yes]:
Do you need LK docker stack? [yes]:

Do you want to define common mgmt nodes for components? [yes]:
Mgmt node 1 IP/hostname: 10.250.10.11
Mgmt node 2 IP/hostname: 10.250.10.12
Mgmt node 3 IP/hostname: 10.250.10.13

Do you want to define common filter nodes for vmagent? [yes]:
Filter node 1 IP/hostname: 10.250.20.11
Filter node 2 IP/hostname: 10.250.20.12
Filter node 3 IP/hostname: 10.250.20.13

>>> Running ClickHouse cluster bootstrap...
=====
ClickHouse cluster bootstrap
=====

ClickHouse Keeper mode: embedded/standalone [embedded]:

ClickHouse nodes source: mgmt/custom [mgmt]:

Writing inventory file:
/home/operator/cresp-onprem-ansible/inventory/clickhouse.ini
Writing group vars:
/home/operator/cresp-onprem-ansible/inventory/group_vars/clickhouse.yml
Writing vault vars:
/home/operator/cresp-onprem-ansible/inventory/group_vars/clickhouse_vault.yml
Writing playbook:
/home/operator/cresp-onprem-ansible/playbooks/deploy-clickhouse.yml
Encrypt ClickHouse vault file:
ansible-vault encrypt
/home/operator/cresp-onprem-ansible/inventory/group_vars/clickhouse_vault.yml

ClickHouse bootstrap completed.

Bootstrap completed.

Vault password file:
/home/operator/cresp-onprem-ansible/.ansible/vault-password

ansible-playbook playbooks/deploy.yml -u {user} --vault-password-file
/home/operator/cresp-onprem-ansible/.ansible/vault-password

```

5.6 Проверка inventory после bootstrap

После завершения bootstrap проверьте список целевых хостов и групп inventory:

```
ansible-inventory --graph
```

Если в выводе отсутствует ожидаемый хост или хост попал не в ту группу, необходимо скорректировать настройки установщика либо повторить bootstrap с правильной топологией.

6 Установка программного обеспечения

6.1 Установка всех выбранных компонентов

Для установки всех компонентов, выбранных во время bootstrap, выполните общий playbook deploy.yml. Команда выполняется из рабочей директории установщика.

```

ansible-playbook playbooks/deploy.yml \
-u <ssh-user> \
--vault-password-file .ansible/vault-password

```

Параметр `-u` задает SSH-пользователя, от имени которого Ansible подключается к целевым хостам. Если для `sudo` требуется пароль, используйте дополнительный параметр `--ask-become-pass`.

```
ansible-playbook playbooks/deploy.yml \  
-u <ssh-user> \  
--ask-become-pass \  
--vault-password-file .ansible/vault-password
```

6.2 Установка отдельных компонентов

Если необходимо установить только один компонент или выполнить установку компонентов поэтапно, используйте соответствующий `playbook`. Ниже приведены типовые команды.

```
ansible-playbook playbooks/deploy-clickhouse.yml -u <ssh-user> --vault-password-file  
.ansible/vault-password  
ansible-playbook playbooks/deploy-kafka.yml -u <ssh-user> --vault-password-file  
.ansible/vault-password  
ansible-playbook playbooks/deploy-victoria-metrics.yml -u <ssh-user>  
ansible-playbook playbooks/deploy-vmagent.yml -u <ssh-user>  
ansible-playbook playbooks/deploy-vector.yml -u <ssh-user> --vault-password-file  
.ansible/vault-password  
ansible-playbook playbooks/deploy-postgres-ha.yml -u <ssh-user> --vault-password-file  
.ansible/vault-password  
ansible-playbook playbooks/deploy-lk.yml -u <ssh-user> --vault-password-file  
.ansible/vault-password
```

Компоненты, использующие `vault`-переменные, запускаются с параметром `--vault-password-file`. Для компонентов, не использующих зашифрованные переменные, этот параметр может не требоваться.

6.3 Повторный запуск установки и обработка ошибок

Ansible `playbook`-и допускают повторный запуск после устранения ошибки. Если команда завершилась ошибкой на одном из хостов, следующие этапы на этом хосте в рамках того же запуска могут быть пропущены. Это является стандартным поведением Ansible.

В таком случае необходимо определить причину ошибки по выводу `playbook`-а, исправить ее и повторно запустить ту же команду. Перед повторным запуском рекомендуется проверить доступность хоста, права `sudo`, доступ к репозиториям и корректность `vault password`.

7 Проверка результата установки

7.1 Проверка системных сервисов

После завершения установки необходимо проверить состояние системных сервисов на целевых хостах. Набор команд зависит от выбранного состава компонентов.

```
systemctl status clickhouse-server  
systemctl status kafka  
systemctl status victoria-metrics  
systemctl status vmagent  
systemctl status vector-cresp-custom@1  
systemctl status vector-cresp-log@1  
systemctl status patroni  
docker ps
```

Сервисы должны находиться в активном состоянии, а контейнеры личного кабинета и API управления должны быть запущены. Если отдельный сервис не запустился, необходимо изучить журнал `systemd`, журналы компонента и вывод Ansible playbook-а.

7.2 Проверка PostgreSQL HA и Patroni

Для проверки состояния Patroni используйте HTTP-endpoint одного из `mgmt`-узлов:

```
curl http://10.250.10.11:8008/cluster
```

В ответе должен отображаться состав кластера и состояние его участников. Для production-стенда необходимо дополнительно убедиться, что определен лидер, реплики доступны, а состояние кластера соответствует выбранной топологии.

7.3 Проверка VictoriaMetrics

Для single-node режима VictoriaMetrics выполните проверку health endpoint:

```
curl http://10.250.10.11:8428/health
```

Для cluster-режима VictoriaMetrics проверьте health endpoint-ы основных компонентов:

```
curl http://10.250.10.11:8480/health
curl http://10.250.10.11:8481/health
curl http://10.250.10.11:8482/health
```

7.4 Проверка контейнерного контура личного кабинета

Для проверки контейнерного контура личного кабинета выполните команду `docker ps` на узлах, где установлен LK docker stack. В списке должны отображаться запущенные контейнеры `frontend`, `backend/API` и смежных сервисов, входящих в состав конкретной версии поставки.

```
docker ps
```

Дополнительно рекомендуется проверить доступность web-интерфейса и API управления с рабочей станции администратора или из разрешенного административного сегмента сети.

7.5 Критерии успешного завершения установки

Установка считается успешно завершенной при выполнении следующих условий:

- общий playbook или playbook-и выбранных компонентов завершились без неустранимых ошибок;
- все установленные системные сервисы находятся в активном состоянии;
- контейнеры LK docker stack запущены;
- PostgreSQL HA/Patroni отображает корректное состояние кластера;
- VictoriaMetrics health endpoint-ы отвечают успешно;
- агенты `vmagent` и `Vector` запущены на предусмотренных узлах;
- административный интерфейс и API управления доступны из разрешенной сети;
- оператор сохранил используемую версию архива, `inventory`, `.installer.env`, журнал установки и сведения о выполненных проверках.

8 Обновление программного обеспечения

8.1 Общий порядок обновления

Обновление Servicepipe ADC выполняется с использованием нового online installer archive. В новом комплекте поставляется файл `.installer.env`, в котором зафиксированы URL репозитория, новые имена архивов и container images.

Общий порядок обновления:

- распаковать новый online archive во временную директорию;
- заменить `.installer.env` в рабочей директории установленного инсталлятора;
- обновить рабочие переменные установщика без переопределения топологии;
- запустить update-команды для всех компонентов или только для выбранного компонента;
- проверить состояние сервисов после обновления.

8.2 Подготовка нового online-архива

Для подготовки `.installer.env` из новой поставки выполните следующие команды:

```
mkdir -p /tmp/cresp-onprem-update
tar -xzf cresp-onprem-ansible-online-<new-version>.tar.gz -C /tmp/cresp-onprem-update
cp /tmp/cresp-onprem-update/cresp-onprem-ansible/.installer.env .installer.env
```

После замены `.installer.env` обновите рабочие переменные установщика:

```
./bootstrap.sh --refresh-installer
```

8.3 Обновление всех компонентов

Для обновления всех компонентов выполните команды по порядку. Если какой-либо компонент не используется на стенде, соответствующую строку необходимо пропустить.

```
ansible-playbook playbooks/update-local-repo.yml \
-u <ssh-user>

ansible-playbook playbooks/update-clickhouse.yml \
-u <ssh-user> \
--vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-kafka.yml \
-u <ssh-user> \
--vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-victoria-metrics.yml \
-u <ssh-user>

ansible-playbook playbooks/update-vmagent.yml \
-u <ssh-user>

ansible-playbook playbooks/update-vector.yml \
-u <ssh-user> \
--vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-postgres-ha.yml \
-u <ssh-user> \
--vault-password-file .ansible/vault-password

ansible-playbook playbooks/update-lk.yml \
-u <ssh-user> \
--vault-password-file .ansible/vault-password
```

8.4 Обновление одного компонента

Для обновления одного компонента выполните только соответствующую update-команду из перечня выше. Перед обновлением отдельного компонента необходимо убедиться, что новая версия совместима с остальными компонентами установленного стенда.

8.5 Обновление только container images личного кабинета

Если необходимо обновить только container images ЛК без изменения остальных настроек личного кабинета, выполните команду:

```
ansible-playbook playbooks/update-lk-images.yml \
-u <ssh-user> \
--vault-password-file .ansible/vault-password
```

После обновления container images необходимо проверить список контейнеров, доступность web-интерфейса и API управления.

9 Удаление программного обеспечения

9.1 Общие правила удаления

Удаление компонентов выполняется отдельными playbook-ами. По умолчанию команда удаления убирает service/unit/config/runtime-файлы компонента, но не удаляет рабочие данные и не удаляет RPM-пакеты.

Примечание. Перед удалением компонентов с данными необходимо убедиться, что выполнено резервное копирование требуемых конфигураций, журналов, баз данных и иных материалов, которые должны быть сохранены.

9.2 Удаление компонентов

Для удаления компонентов используйте соответствующие playbook-и:

```
ansible-playbook playbooks/remove-lk.yml -u <ssh-user>
ansible-playbook playbooks/remove-vector.yml -u <ssh-user>
ansible-playbook playbooks/remove-vmagent.yml -u <ssh-user>
ansible-playbook playbooks/remove-postgres-ha.yml -u <ssh-user>
ansible-playbook playbooks/remove-kafka.yml -u <ssh-user>
ansible-playbook playbooks/remove-clickhouse.yml -u <ssh-user>
ansible-playbook playbooks/remove-victoria-metrics.yml -u <ssh-user>
ansible-playbook playbooks/remove-local-repo.yml -u <ssh-user>
```

9.3 Удаление с данными и пакетами

Для удаления компонента вместе с рабочими данными используйте переменную `component_purge_data=true`:

```
ansible-playbook playbooks/remove-kafka.yml \
-u <ssh-user> \
-e component_purge_data=true
```

Для удаления компонента вместе с RPM-пакетами используйте переменную `component_remove_packages=true`:

```
ansible-playbook playbooks/remove-kafka.yml \
-u <ssh-user> \
-e component_remove_packages=true
```

Для полного удаления компонента вместе с рабочими данными и пакетами используйте обе переменные:

```
ansible-playbook playbooks/remove-kafka.yml \
-u <ssh-user> \
-e component_purge_data=true \
-e component_remove_packages=true
```

9.4 Рекомендуемый порядок удаления

При полном удалении стенда рекомендуется удалять компоненты в следующем порядке:

- LK;
- Vector;
- vmagent;
- PostgreSQL HA;
- Kafka;
- ClickHouse;
- VictoriaMetrics;
- local-repo client configuration.

Такой порядок позволяет сначала остановить пользовательский и управляющий контур, затем агенты доставки данных, после этого базы данных, брокеры событий, аналитические хранилища и настройки репозитория.

10 Типовые проблемы и способы устранения

10.1 Для sudo требуется пароль

Если при выполнении playbook-а появляется сообщение `sudo: a password is required`, значит SSH-пользователь не может выполнить `sudo` без пароля. Запустите команду с параметром `--ask-become-pass` или настройте `sudo` на целевых хостах.

```
ansible-playbook playbooks/deploy.yml \
-u <ssh-user> \
--ask-become-pass \
--vault-password-file .ansible/vault-password
```

10.2 Хост завершился с ошибкой, следующие этапы пропущены

Если хост получил `failed=1`, следующие этапы на этом хосте в рамках того же запуска могут быть пропущены. Это стандартное поведение Ansible. Необходимо исправить причину ошибки и перезапустить команду.

10.3 DNF не находит пакет

Если DNF не находит требуемый пакет, проверьте доступность online RPM repository с целевой ноды и наличие пакета в репозитории.

```
dnf clean all
dnf makecache
dnf repoquery <package-name>
```

10.4 Не скачивается архив из raw repository

Если не скачивается archive или иной файл из raw repository, проверьте URL raw repository, указанный при подготовке установщика, и доступность требуемого файла.

```
curl -I <raw-repository-url>/<file-name>
```

10.5 Неверный vault password

Если Ansible сообщает об ошибке расшифровки vault-файлов, убедитесь, что используется тот же vault password, который вводился во время bootstrap. Проверку можно выполнить командой:

```
ansible-vault view inventory/group_vars/lk_nodes_vault.yml \
--vault-password-file .ansible/vault-password
```

Если файл `.ansible/vault-password` был заменен или поврежден, необходимо восстановить корректный файл пароля из защищенного хранилища либо повторно сформировать vault-файлы в установленном порядке.

11 Завершение работ и хранение установочных материалов

После завершения установки, обновления или удаления необходимо сохранить сведения, позволяющие подтвердить состав выполненных работ и воспроизвести конфигурацию стенда.

Рекомендуется сохранить:

- версию online installer archive и его контрольную сумму;
- использованный release manifest и файл `.installer.env`;
- `inventory` и `group vars`, сформированные bootstrap;
- сведения о выбранной топологии и ролях узлов;
- перечень запущенных playbook-ов и их результат;
- журнал выполнения установки или обновления;
- результаты проверок `systemctl`, `docker ps`, `health endpoint`-ов и доступности административного интерфейса;
- сведения об ответственных администраторах и дате выполнения работ.

Файл `.ansible/vault-password` и иные секреты должны храниться отдельно от общедоступной документации и журналов работ. Доступ к ним должен предоставляться только уполномоченным сотрудникам, ответственным за сопровождение установленного экземпляра Servicepipe ADC.